A Holistic Approach to Undesired Content Detection in the Real World

Warning: some content may contain racism, sexuality or other harmful language.

Todor Markov Chong Zhang Sandhini Agarwal Tyna Eloundou
Teddy Lee Steven Adler Angela Jiang Lilian Weng
OpenAI

Abstract

We present a holistic approach to building and deploying a reliable and robust undesired content detection model into the real world for content moderation purposes. The success of such a system relies on a chain of carefully designed and executed steps, including taxonomy and labeling instruction design, data quality control, active learning pipeline to capture rare events, as well as a variety of methods to robustify the model to improve performance and avoid overfitting. Our moderation system is trained to detect several categories of undesired content, including in the areas of sexual content, hateful content, violence, self-harm, and harassment. Our approach is compatible with a very wide range of different content taxonomies, and it can be used to create high-quality content classifiers that outperform off-the-shelf models.

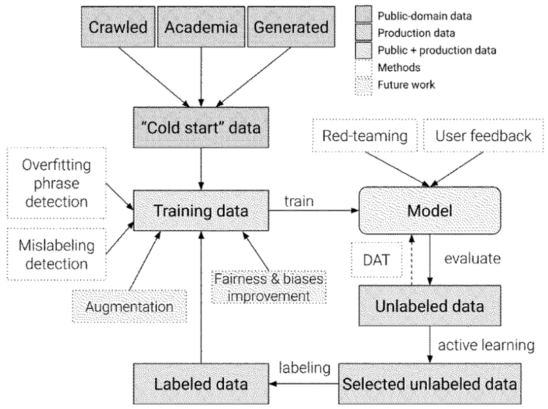

Figure 1: Overview of the model training framework.

1 Introduction

Recent advancements in deep learning enable us to build AI systems that are powerful enough to be deployed to benefit some socioeconomic tasks in the real world (Silver et al., 2018; Devlin et al., 2019; Andrychowicz et al., 2020; Brown et al., 2020; Cohen et al., 2022). To responsibly deploy generative language models in the world, we need to ensure several safety conditions or control over the models. First, model providers would like assurances that the models will not produce content that is disallowed by their policies. Second, customers of these models sometimes require control over content to mitigate the impact of sensitive user cases and reduce brand risk. A principled, robust, and efficient moderation solution can track and measure the model inputs and outputs to ensure safety standards, and to provide fine-grained control to enable use cases with sensitive needs, such as educational applications. We also believe a strong undesired content classifier lays the foundation for building safer AI systems in the wild, as it enables the capacity of moderating, evaluating, and guiding the models towards safer behavior.

Existing work on content detection focuses mainly on a limited set of categories, including toxicity (Pavlopoulos et al., 2020; Gehman et al., 2020), hate speech (Kwok and Wang, 2013; Davidson et al., 2017), and abusive content (Nobata et al., 2016; Vidgen et al., 2019); or is tailored towards a targeted use case, such as Perspective API (Jigsaw, a) on online toxic comment moderation. There is increasing attention to understanding the risk areas of large language models via a more rigorous taxonomy (Weidinger et al., 2021), but the amount of work is still limited, especially for deploying language models for the real world applications. Here we aim to build a more comprehensive system for detecting a broad set of categories of undesired content, including sexual content, hateful content, violence, self-harm, and harassment, as well as severe subcategories under each top-level category.

Detecting undesired content is difficult due to several challenges. First, there is not a clearly and widely agreed-upon categorization of undesired content. Designing a detailed taxonomy for potentially unsafe content and operationalizing it for labeling purposes require a lot of work. There exist a significant number of corner cases that need to be clarified further within the categorization framework to achieve a high enough agreement during labeling. In addition, labeling decisions are subjective due to the different social and cultural backgrounds of human annotators. Second, a practical moderation system needs to process real-world traffic. Thus a model bootstrapped from public domain data or academic datasets would not work well because there exists a big data distribution shift and taxonomy misalignment. Third, it is rare to encounter certain categories of undesired content; For example, we only observed 0.04% self-harm and 0.017% hateful content involving threats in the traffic of our production system. Hence, we need smart solutions to the cold start problem and effective ways to discover positive samples in time.

Multiple components contribute to the success of building and deploying a practical, general moderation system into the real world, effectively establishing a chain of carefully polished and curated configurations for data collection, data labeling, model training and active learning. On the basis of our experimentation, we find the following conclusions to be especially noteworthy.

- *Data quality is the key.* However, data labeling is difficult even when working with well-trained and highly-skilled labelers. Both the sophistication of the labeling instruction and a set of properly designed quality metrics contribute significantly to the quality of final outcomes. A poorly chosen quality metric may lead to data that hurts performance. Furthermore, there is inherent subjectivity in definitions of categories for sensitive content which may differ between people and groups. (See §3.2).

- *Active learning is a necessity.* There is likely a large distribution shift between the public domain data and the traffic from one's production system. Thus, it is critical to collect new training samples from the production traffic. Active learning can effectively expand the training dataset to capture a meaningful amount of positive samples when dealing with exceptionally rare events and adapting to the data distribution shift in time. (See §3.1)

- *Use public datasets with care.* Publicly available data might not lead to high quality performance for the problem in hand due to different taxonomy and training data distribution, but can be used to construct a noisy cold start dataset at the early stage. However, adding academia data into the training set may hurt the model performance at a later stage when there are enough properly labeled data samples.

- *Be aware of overfitting.* Deep learning models could easily overfit common phrases or templates. For example, the model can over-generalize to anything formatted as `"X is hateful"` if the data distribution is off-balance. We tackle this challenge by programmatically identifying overfitting phrases and by red-teaming via human trials. We then alter the training distribution by incorporating model-generated or human-curated synthetic data to patch the weakness. (See §3.5 and §3.3)

- *Mistakes in data will happen and need to be managed.* Even with a significant amount of efforts on data quality control, we still run into mislabeled examples. We explore different methods for identifying those cases, including cross-validation and hunting down overfit common phrases via token subtraction. (See §3.2 and §3.5)

We aim to present a holistic approach to building a reliable and robust undesired content detection model for real-world applications. Our approach is meant to handle the scenario in which the type of the content to be detected is rarely observed in the real world. We hope that the lessons we learned are useful to others working on similar problems. We release a dataset [1] containing text samples from the public domain labeled according to our taxonomy. The model is accessible via OpenAI API Moderation endpoint [2].

2 Taxonomy

Given that the meaning of a sentence can depend on context, designing a universal taxonomy for undesired content is challenging and requires inherent trade-offs. Throughout this exercise, we aimed

[1] `https://github.com/openai/moderation-api-release`

[2] `https://beta.openai.com/docs/guides/moderation`; Harassment category is currently under some improvement and will be available soon.

to make the dataset as broadly applicable as possible to maximize its utility across research and industrial contexts, but recognize it still ought to be critically evaluated prior to use in any context, The anatomy of what's undesired can vary based on factors like use cases, users, regions and many other factors. Some sentences labeled as undesired according to our taxonomy (e.g. "`I will kill you`" would be labeled as positive for violence) may not be considered undesired when appearing in another context such as fiction.

Additionally, we aimed to capture layers of subcategorization and granularity. Much existing work lacks subcategorizations to capture a spectrum of severity, which is critical, since statements such as "`kill all [IDENT_GROUP]`" can have different repercussions compared to "`[IDENT_GROUP] are dishonest`". The existence of subcategories can also largely reduce the disagreement rate and corner cases during labeling, making it valuable for data quality control. In the following outline, we present our taxonomy for undesired content categorization, including five top-level categories and a spectrum of subcategories.

`S`: **Sexual content** Content that depicts explicit or implicit sexual behavior. Subcategories that are considered undesired: `[S3]` sexual content involving minors, `[S2]` content that depicts certain sexual activities which could be illegal[3] if they happened in real life today, and `[S1]` erotic sexual content that does not depict illegal activities. Subcategories that are not considered undesired: `[S0]` Non-erotic or contextualized sexual content, such as medical or sex education material.

`H`: **Hateful content** Content that is threatening, insulting, derogatory, and otherwise abusive content targeting specific chosen groups or members of the group because of their group identities. Subcategories that are considered undesired: `[H2]` hateful content calling for violence or threatening, `[H1]` derogatory stereotypes or support for hateful statements. Subcategories that are not considered undesired: `[H0.a]` neutral statement referring to group identity and `[H0.b]` contextualized hate speech, such as a quote of other individual's statement in order to provide commentary.

`V`: **Violence** Content that depicts or shows support for physical violence. Subcategories that are considered undesired: `[V2]` extremely graphic violence, `[V1]` threats or support for violence. Subcategories that are not considered undesired: `[V0]` neutral depictions of contextualized violence.

`SH`: **Self-harm** Content that contains a graphic description or shows support for self-harm.

`HR`: **Harassment** Content that may be used to torment or annoy individuals in real life, or make harassment more likely to occur.

Our model learns to predict whether a given sample violates any of *8 chosen categories*, including all the top categories (`S`, `H`, `V`, `SH`, `HR`) and three most severe subcategories (`S3`, `H2`, and `V2`).

3 Methods

3.1 Data Selection and Active Learning

To ensure that our moderation system performs well in the context of our production use cases, we need to incorporate production data to our training set. We take a three-stage procedure in an iterative fashion.

First, a large volume of our production data is selected at random, in which any potential personally identifiable information (PII) is masked. The most recent moderation model is used to score these samples and discover which ones may trigger any chosen categories.

In the second stage we run a simple active learning strategy to select a subset of most valuable samples to be labeled out of the random samples extracted in stage one. The active learning strategy is made up of three parallel pipelines. The first one relies on random sampling such that some fraction of our data remain consistent with the underlying data distribution in production. The second one randomly selects from samples with model score above a certain probability threshold for every category to identify likely positive data points. The last pipeline adopts a set of uncertainty sampling strategies (Lewis and Gale, 1994; Lewis and Catlett, 1994) to capture samples that the model is most uncertain about, which means that the predicted probability for that category is close to 50%.

During the final stage, all the samples selected by different active learning strategies are aggregated and re-weighted based on the model used

[3] This mapped most closely to what's illegal in USA.

to generate the sample and certain other metadata associated with it. The weight is configured to be the square root of the sample count. This helps improve the diversity of selected samples with regard to the associated metadata. We update the sub-strategy mixture over time based on changes in the data distribution and categories that we want to improve the most at different stages.

3.2 Labeling and Quality Control

Data label correctness is critical to good model performance. Getting such data can be difficult given that our categories and the boundary lines between them are inherently subjective. However, certain interventions can significantly improve the quality of labeled data.

One important intervention for improving data quality - in terms of both consistent labels across different labelers as well as between labelers and researchers - is to make the labeling instructions as *well-defined* and *concrete* as possible. To make the instructions well-defined, we sought to provide detailed definitions and design categories or subcategories to be as mutually exclusive as possible so as to minimize ambiguity. To make the instructions concrete, we provide numerous examples are each category. We also host regular calibration sessions to review ambiguous edge cases and instances where external labelers and our internal auditors disagree. Based on feedback from those sessions, we make the instructions more clear and concrete, which helps improve labeling accuracy.

Regular, ongoing audits are necessary to ensure that labeled data is consistently in a sufficiently high quality. The choice of which samples to audit and what metrics to use to measure data quality is crucial. We found that selecting auditing targets at random cannot maximize the value out of auditing due to the imbalanced distribution across categories. The labeler-auditor agreement rate (i.e. accuracy) is suboptimal because positive examples are rare events to encounter and the accuracy can be arbitrarily high due to the abundance of true negatives. Instead, in each chosen category, we randomly select 10 samples with positive labels and 10 samples with model probability greater than 50%. The former help capture false positive cases and the latter provide an estimation on recall. Then we compute the F-1 score for the chosen samples based on the annotator-assigned labels while using auditor-assigned labels as ground truth. This procedure performs much better in practice when certain categories of undesired data points are rare. Separation of metrics per category makes it easy to recognize category-specific issues and allocate focus to weaker performers.

Even with very clear labeling instructions and an effective audit procedure, mistakes in data are still unavoidable. To identify potentially mislabeled samples in our dataset, we periodically split our current training dataset into two parts, train separate models on those datasets and use each model to score another half of the dataset that model was not trained on. When the model prediction disagrees with the current ground-truth label, the sample in question gets flagged. A random portion of flagged samples is audited, and if more than 30% are identified as mislabeled, all flagged samples would get labeled again for the second time.

3.3 Synthetic Data

On top of the data collection discussed above, we also use synthetic data to improve the model performance on rare categories such as `SH` and `V2` and mitigate the bias towards demographic attributes. Synthetic data generated by large pre-trained language models has been shown to be an effective way for data augmentation (Anaby-Tavor et al., 2020; Kumar et al., 2020) and it is particularly helpful when there is little to no initial data ("cold start") or when there are not enough positive samples in the production traffic.

Zero-shot data for cold start. To kick start the active learning and labeling process, we need some initial data to build the first version of the model and train labelers. However, it is difficult to find existing public datasets on certain categories such as `SH` and `V2`. We tackle the problem by generating a synthetic dataset with zero-shot prompts on GPT-3. The prompts are constructed from human-crafted templates and we label the generated texts as the initial dataset. Table 1 provides an example prompt for `SH`.

Few-shot data for rare categories. After running the active learning pipeline for a few iterations, we found that the model still suffers from poor performance on categories such as `SH` and `V2`, since both are extremely rare events in our production traffic and the model is not getting enough learning signals. To collect more positive data, we construct few-shot prompts with existing positive examples and send the generated texts to be labeled. We

Example prompt
>>> The text is about self harm, more specifically: physical self-harm (cutting).
>>> The narrator is third person (a friend).
>>> The narrator is a young teenager (female).
>>> The act of self harm happens: last week.
>>> Type of text: online forum post.
>>> The following word should come up in the text: camera.
>>> The text should not contain the phrase "self harm" or "self-harm".
>>> The writing level is: average, some errors.
>>> The text starts below:

Table 1: Example zero-shot prompt template for generating synthetic `SH` data. The sections in green are filled with random ingredients to encourage diversity.

observed a nontrivial performance improvement by incorporating the synthetic dataset.

Curated data to mitigate bias. Similar to other existing NLP models, our models also suffer from bias towards certain demographic attributes as bias commonly exists in data. We mitigate the issue by curating a synthetic dataset with templates that tend to lead to hateful predictions, e.g., "`[subject] is selfish/foolish/narrow-minded.`". The `[subject]` could either be filled with real demographic attributes (e.g., `Latino`) or random object names (e.g., `black blanket`), the former is a hateful sample and the latter is innocent. We observe that the curated dataset not only mitigates bias to some degree, but also helps improve the model performance. For instance, the average AUPRC on hateful content was improved from 0.417 to 0.551 by adding 69k curated synthetic examples. We believe this is because the contrastive setup of subjects in synthetic example templates encourages the model to infer the correct feature representations: negative descriptive words are not enough to be considered hate, but only when applied to protected identity groups.

Large amount of noisy data does not help. To understand whether it is helpful to include a large amount of noisy synthetic data, we also generated zero-shot and few-shot examples twice the size of the existing labeled training dataset. For zero-shot examples, we set the label to positive or negative if the prompt asks the model to generate undesired or safe examples, respectively. For few-shot examples, we set the label to positive or negative if all of the few-shot examples are positive or negative, respectively. Contradictory to previous studies (Wang et al., 2021; Schick and Schütze, 2021), we found it hurt the model performance by mixing the noisy synthetic data into training. It is worth noting that many existing studies on synthetic data usage experimented in the no-to-low data regime, where only a handful of labels are available. However, in our experiment, we have collected a decent size of high-quality data labels and we suspect that noise introduced by synthetic data confuses the model and lowers the learning efficiency.

3.4 Domain Adversarial Training

We intended to make good use of existing public NLP datasets to improve the performance of our models. However, we observed that models trained on public NLP datasets do not perform well on our production traffic. This is likely due to the distribution difference between domains. For instance, examples from our production traffic are usually much longer and contain few-shot prompts, whereas the existing public NLP datasets are usually shorter and often crawled from Wikipedia, Twitter, etc. (Vidgen and Derczynski, 2020). To mitigate the problem, besides carefully tuning the mixture of public datasets and production data, we in addition apply Wasserstein Distance Guided Domain Adversarial Training (WDAT) to encourage the model to learn domain invariant representations (Arjovsky et al., 2017; Ganin et al., 2016).

We follow Shen et al. (2018) and approximate the Wasserstein distance by maximizing the loss of a domain critic head. Let $f_z(x) : \mathbb{R}^d \to \mathbb{R}^z$ be the feature extractor that maps the d-dimensional input into a z-dimensional embedding, $f_c(h) : \mathbb{R}^z \to \mathbb{R}^c$ be a multiclass classification head, and $f_d(h) : \mathbb{R}^z \to \mathbb{R}$ be the domain critic head that maps the embedding into real number. The domain critic loss is defined as

$$\mathcal{L}_d(\mathcal{D}_s, \mathcal{D}_t) = |\mathop{\mathbb{E}}_{x \in \mathcal{D}_s} f_d(f_z(x)) - \mathop{\mathbb{E}}_{x \in \mathcal{D}_t} f_d(f_z(x))|.$$

Combined with the regular classification loss $\mathcal{L}_c$, our objective is to solve the following minimax problem:

$$\min_{\theta_z, \theta_c} \{\mathcal{L}_c + \lambda \max_{\theta_d} \mathcal{L}_d\},$$

where $\theta_z, \theta_c, \theta_d$ are the parameters for f_z, f_c, f_d, respectively. Our model uses a transformer encoder as the feature extractor f_z.

In our implementation, we use the absolute value in $\mathcal{L}_d$ since the initial loss could be negative, and clip θ_d in a compact space $[0.01, 0.01]$ to enforce the Lipchitz constraint. We empirically set the balancing coefficient λ to 0.01. In experiments, WDAT achieves a more stable training compared to the original classifier-based approach (Arjovsky et al., 2017), and yields better performance on our production traffic with and without labeled production data in the training set.

3.5 Model Probing

It is widely known that ML models are vulnerable to adversarial inputs and may make predictions based on seemingly irrelevant features (Szegedy et al., 2013; Goodfellow et al., 2015; Alzantot et al., 2018; Zhang et al., 2021). For instance, a sentiment classification model may make different predictions for “`a short and moving film.`” and “`a short and moving movie.`”, indicating that the model makes predictions based on `film`/`movie` rather than descriptive words. To make sure that our model performs as expected, we conduct tool-assisted model probing and manual red-teaming.

Key Tokens Probing. Our first goal is to verify whether the model is making predictions based on correct features. For instance, we observed that our models may incorrectly predict “`all immigrants.`” as `hate`, which is possibly because among 21 training examples containing the phrase (from Röttger et al. (2021)), 17 examples are positive and only 4 examples are negative. Table 2 provides a few examples, where the expected behavior is to also take `k1ll` and `D3ath` into account. To identify such potentially overfitted key tokens, we apply input reduction (Feng et al., 2018) on the training dataset by removing tokens greedily until the prediction drops below 0.8. On average, we were able to reduce the number of characters per sample from 722.3 to 15.9. By cross-validating with third-party APIs and heuristic keywords, we infer that about 97% of the identified key tokens are indeed unsafe, indicating that the model is picking up the correct feature most of the time. For the remaining 3% key tokens, we manually reviewed them and discovered the following problems:

1. The frequency of irrelevant symbols is very high (e.g., the period symbol contributes to the prediction score).

2. Some key tokens are very long and close to the original example, hinting at an overfit, as the model memorizes that training sample.

3. Our model is not strong at handling typos and adversarial inputs (e.g., “`w h o r e s`” becomes “`e s`”), which we plan to resolve with better data augmentation as part of future work soon (See §6).

Hate	Example
1	`I want to k1ll all immigrants.`
1	`Why is it that all immigrants are repulsive?`
1	`D3ath to all immigrants.`
0	`Threatening to kill all immigrants will get you thrown in jail.`
...	

Table 2: Example training data from (Röttger et al., 2021) containing the phrase “`all immigrants.`” The model trained on it directly may make predictions based on incorrect features.

To address the above issues, we manually label the remaining key tokens and add them back to the training dataset.

Human Red-teaming. As a final layer of the verification process, we conduct internal red-teaming before releasing new models. The goal of the red-teaming is to uncover any unexpected model weaknesses beyond the test dataset. During the process, we discovered the following issues:

1. The model overfits on the # tokens on the `hate` category, which is likely due to the fact that we have many `hate` training examples from some academia datasets containing only short tweets.

2. The model overfits racial tokens such as `black`. e.g., “`I hate black people!`” should trigger `hate` but “`I hate black cats!`” should not.

To mitigate the above issues, we construct synthetic datasets from hand-curated templates and synthetic model generations to patch the holes, and adjust the training dataset distribution to make sure we have the right mix across multiple types of text sourced from academia datasets. The process can be iterated, helping us discover new issues and solutions in each round and naturally leading to improved robustness and consistency in time when the red-teaming process can be executed more regularly and at scale.

4 Experiment Results

4.1 Model Architecture and Training

Our model is a light weight transformer decoder model where the final output linear layer is replaced with 8 MLP heads, each corresponding to one independent matrix of shape $[d_{\text{model}}, 256, 1]$, where d_{model} is the transformer model size. We find this head architecture works better than a single deep MLP layer with one output vector of 8 dimensions at avoiding interference between categories and requires fewer parameters to train.

The model is initialized from a pre-trained GPT-3 model of the same size and fine-tuned with learning rate 0.05, batch size 256, dropout rate 0.1 within MLP heads and up to 3 epochs.

4.2 Model Performance

Our model is trained and tested on both production and public domain data. We are not able to share the test dataset containing production traffic for privacy and legal reasons; hence, we report the model performance on a different test dataset[4] containing only samples from the public domain, as well as several publicly available datasets on undesired content detection.

Table 3 compares the performance of our model with Perspective API[5] as a baseline on our test dataset, TweetEval (Barbieri et al., 2020), Stormfront hate speech dataset (de Gibert et al., 2018), a subset of Reddit comments with noisy labels on erotic content processed according to Barrientos et al. (2020) and a downsampled Jigsaw toxic comments test dataset (Jigsaw, b). None of the training portion of external evaluation benchmarks are incorporated into our training, except for half of Jigsaw's training data that has no overlap with the Jigsaw test set in evaluation. Unfortunately, due to the taxonomy mismatch, we cannot have exact comparison across all categories. For example, our taxonomy does not cover "toxic" and Perspective API does not explicitly detect "self-harm" or "sexual content". See the details on how we match two taxonomies and preprocess each test dataset in Appendix. A.

It is not surprising that our model performs the best on the test dataset labeled with same taxonomy and the Perspective API does a better job on Jigsaw data. It further proves the point on how important it is to align the taxonomy between training data and use cases in evaluation. Our model outperforms the Perspective API baseline on both TweetEval and Stormfront test sets for detecting hateful content, despite the fact that neither are in the training set.

4.3 Active Learning Experiments

To assess the importance of active learning, we evaluate the performance of our active learning strategy, as described in §3.1, compared to random sampling.

		Perspective	Ours
Public	`S`	0.8709*	0.9703
	`H`	0.6914	**0.7968**
	`V`	0.5201	**0.7371**
	`HR`	0.3902*	0.6191
	`SH`	-	0.8070
	`S3`	-	0.7638
	`H2`	-	0.7268
	`V2`	-	0.6061
Jigsaw	Identity-hate	0.6644	**0.6890**
	Insult	**0.8814**	0.8548
	Obscene	0.9500	0.8353*
	Threat	**0.7492**	0.6144
	Toxic	0.9769	0.9304*
TweetEval	Hate	0.5961	**0.6473**
	Offensive	0.7919*	0.7024*
Stormfront	Hate	0.8754	**0.9053**
Reddit	Sexual	0.8961*	0.9417*

Table 3: Comparison of our model with Perspective API on AUPRC (Area under the Precision-Recall Curve) across a set of test datasets. Numbers followed with "*" are based on *approximated* taxonomy match, so not an exact fair comparison.

Iterative training We run the following training procedure twice, using our active learning strategy and random sampling, respectively.

1. Start with an initial training dataset $\mathcal{D}_0$ of $k_0 = 6000$ labeled examples from the public domain and a validation set $\mathcal{V}$ of about 5500 samples from the production traffic.

2. for $i \leftarrow 0$ to $N - 1$ do ($N = 3$):

   (a) Train a new model M_i on $\mathcal{D}_i$;

   (b) Evaluate M_i on $\mathcal{V}$;

   (c) Score 5×10^5 production samples with M_i from our production traffic;

   (d) Choose about 2000 samples from the above data pool via the selection strategy in test and add samples to the training set to construct $\mathcal{D}_{i+1}$ after labeling.

Results Table 4 demonstrates the label distributions obtained by the two strategies and our active learning strategy can capture undesired content 10+ times more effectively than random sampling on all categories. Overall about 40% of samples selected by active learning can trigger at least one positive

[4] `https://github.com/openai/moderation-api-release`

[5] `https://www.perspectiveapi.com/`

Category	Random Sampling	Active Learning	Multiplier
S	1.49%	25.53%	17.1×
H	0.17%	3.09%	18.2×
V	0.48%	9.92%	20.7×
HR	0.55%	6.41%	11.7×
SH	0.09%	1.85%	20.6×
S3	0.24%	2.42%	10.1×
H2	0.03%	0.67%	22.3×
V2	0.25%	4.27%	17.1×
Safe	96.57%	59.54%	-

Table 4: Label distributions for samples selected by random sampling and active learning sampling. Note that one sample can be assigned with multiple labels so the percentages sum up to more than 100%.

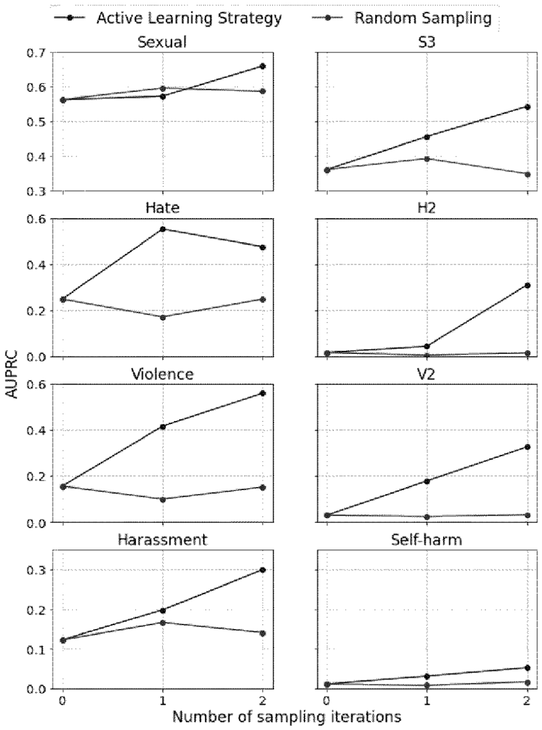

Figure 2: Performance of active learning sampling versus random sampling on the same validation set at each model iteration, measured by AUPRC.

label, while in comparison only 3.4% of random samples are assigned with any positive label.

As shown in Fig. 2, using the active learning strategy to decide which new data samples leads to a greater improvement across all categories than random sampling. We observe significant performance improvement on the majority of the categories with active learning after only 2 iterations. AUPRC for SH increases from 1.6% to 5.2% but it still remains weak. We hypothesize that it is because positive SH samples are especially rare compared to other categories, and we believe after a few more rounds of iterations, the performance should grow much stronger.

4.4 Domain Adversarial Training Experiments

We want to understand the effectiveness of Wasserstein Distance Guided Domain Adversarial Training (WDAT) under three scenarios: (1) At the beginning of the project, we only have labeled public data and unlabeled production data. (2) In the middle of the project, we also curate synthetic examples to improve model weaknesses. (3) At the later stage, we get a sufficient amount of labeled production examples. All three circumstances are important because we want to make good use of unlabeled production data to train the best model throughout the project, and a strong model on production traffic boosts the effectiveness of active learning at every iteration. We use the following setup to compare the performance on our production traffic.

Datasets We create three training datasets PUB, SYN, and MIX to study (1), (2), and (3), respectively. PUB consists of around 90k public examples including both samples from academia datasets and Web data (Common Crawl) labeled by our annotators. SYN adds additional 69k curated synthetic examples. MIX contains all examples in SYN with additional 60k production samples with labels.

Models The baseline models are trained with vanilla supervised learning. The DAT models are trained with two hidden layers of 300 dimensions using additional 100k unlabeled production data points. All models are trained with up to 2 epochs, and the training is repeated 3 times with different random seeds.

Results We compare the average AUPRC on the production validation set $\mathcal{V}$. As demonstrated in Table 5, the improvement from WDAT is significant when we only have access to public datasets (PUB), and the marginal gain reduces gradually as we add more training examples, especially in-distribution production samples. For instance, DAT improved SH AUPRC from 0.063 to 0.281 on PUB and from 0.086 to 0.296 on SYN, whereas the improvement is only from 0.621 to 0.632 on MIX. WDAT still helps weak categories (SH and V2) on SYN and MIX, but it may slightly hurt the performance for categories with enough amount of in-distribution

Category	PUB		SYN		MIX	
	Baseline	AUC	Baseline	DAT	Baseline	DAT
S	.698	**.730**	.726	**.745**	**.943**	.939
H	.417	**.491**	**.551**	.476	**.843**	.818
V	.490	**.529**	**.532**	.531	**.640**	.633
HR	.258	**.369**	.326	**.356**	.453	**.482**
SH	.063	**.281**	.086	**.296**	.621	**.632**
S3	.592	**.759**	**.779**	.777	.911	**.936**
H2	.393	**.643**	.570	**.577**	.851	**.854**
V2	.165	**.453**	.093	**.507**	.443	**.533**

Table 5: The average AUPRC on a production validation set. PUB denotes models trained on labeled public datasets, SYN adds additional synthetic examples, and MIX adds additional labeled production examples. We mark the best result within each configuration in **bold**.

data such as H and V. Further study is required to improve the performance on all categories with unlabeled data throughout different stages of the project.

5 Related Work

There is a long track record of work on the definition and detection of hateful, toxic, offensive and abusive content (Kwok and Wang, 2013; Nobata et al., 2016; Waseem, 2016; de Gibert et al., 2018; Vidgen et al., 2019; Gehman et al., 2020; Rosenthal et al., 2020; Lees et al., 2022). Zampieri et al. (2019) proposed a three-level hierarchical taxonomy considering whether the given language is (i) offensive or not; (ii) targeted or not; and (iii) targeted at group, individual, or other organizations. Usually, hateful expression targeting protected identity groups is considered hate speech (Davidson et al., 2017). Perspective API defines toxicity as "A rude, disrespectful, or unreasonable comment that is likely to make people leave a discussion". Some also used toxicity as a general umbrella term for offensive, abusive, and hateful language (Pavlopoulos et al., 2020). The definitions of hatefulness, toxicity, offensiveness, and abusiveness have overlaps but are not exactly the same, creating obstacles for sharing datasets between projects. Furthermore, only a limited amount of work considered detailed subcategorizations (Mollas et al., 2020; Borkan et al., 2019) to capture a spectrum of severity, making it harder to control labeling quality. Finally, there exist various types of potentially undesired text in the wild, such as sexual content involving minors, extreme graphic violence, or support for self-harm or suicides, besides offensive and abusive language, and we observed a gap between current research work and the entirety of content types that should be moderated and detected. Our work aims to fill in the gap.

Despite the common belief that training data quality is critical for model performance, there is still lack of community standards for labeling standards, labeler training, quality metrics, etc. (Vidgen and Derczynski, 2020; Yin and Zubiaga, 2021; Lees et al., 2022; PAI, 2021). Vidgen and Derczynski (2020) studied 60+ datasets for abusive language detection and found that the primary data source is Twitter and expert coding is the most common way to annotation data, closely followed by crowdsourcing. For large-scale data collection, crowdsourcing remains the most common approach (Mollas et al., 2020; Zampieri et al., 2019; Davidson et al., 2017). However, the weak skill set of non-expert annotators can lead to lower data quality (Waseem, 2016; Yin and Zubiaga, 2021). Some recent work turns to large pre-trained language models to generate synthetic data, significantly reducing the cost of time and human labor (wan; Hartvigsen et al., 2022), but it is unclear whether model outputs would be diverse enough to adapt to the real-world distribution. Synthetic data can be hand-crafted (Röttger et al., 2021), but it is limited by size and thus more suitable for evaluation. It is noteworthy that training data can contain bias due to the subjectivity and biases in the data collection process (Davidson et al., 2019; Sap et al., 2019).

Active learning has been successfully applied to a number of different domains such as text classification (Lewis and Gale, 1994; Schohn and Cohn, 2000; Siddhant and Lipton, 2018); machine translation (Zeng et al., 2019); image classification (Luo et al., 2005; Hoi et al., 2006; Gal et al., 2017); object detection (Schmidt et al., 2020) and information retrieval (Shen and Zhai, 2005). There are several families of active learning sampling strategies that are often used in practice. Uncertainty sampling selects data points about which the model is most uncertain. The uncertainty of the model can be quantified by predicted probabilities (Lewis and Gale, 1994; Lewis and Catlett, 1994; Culotta and McCallum, 2005; Scheffer et al., 2001), disagreement among an ensemble of models (Seung et al., 1992; Dagan and Engelson, 1995; McCallum and Nigam, 1998), or by using dropout and Bayesian approaches (Gal et al., 2017; Siddhant and Lipton, 2018). Diversity sampling chooses samples in a way that ensures sufficient diversity within the selection. This is commonly achieved by clustering unlabeled data and sampling from different clusters (Nguyen and Smeulders, 2004; Xu et al., 2007), or by selecting samples which are "representative" of the sample distribution (i.e., which are similar to many other samples) (McCallum and Nigam, 1998; Settles and Craven, 2008). Uncertainty and diversity sampling are sometimes combined in a single complex active learning strategy.

Red-teaming is a common approach for model improvement by discovering and patching the weakness iteratively (Dinan et al., 2019; Vidgen et al., 2020; Kiela et al., 2021; Ziegler et al., 2022; Perez et al., 2022), where humans are encouraged to look for examples that could fail the model. Dynabench (Kiela et al., 2021) is built as a platform for easy adversarial data collection. Mishkin et al. (2022) describes in detail an operational process for doing red-teaming using external experts. Ziegler et al. (2022) designed a tool to efficiently assist human adversaries to identify failures in a classifier. Models trained with red-teaming data are found to be more robust to adversarial attack (Dinan et al., 2019; Ziegler et al., 2022) and human-in-the-loop dynamic data collection can efficiently improve model performance (Kiela et al., 2021; Vidgen et al., 2020).

Domain adaptation aims at generalizing knowledge learned in the source domain towards a related target domain (Ben-David et al., 2006; Weiss et al., 2016; Ben-David et al., 2009), the technique is most useful when there is insufficient labeled data in the target domain but sufficient labeled data in the source domain. Different methods have been proposed to transfer the knowledge across domains (Ramponi and Plank, 2020; Blitzer et al., 2006; Mansour et al., 2008). Inspired by generative adversarial nets (GANs) (Goodfellow et al., 2014) which trains a discriminator to make the representations of source and target indistinguishable, Domain Adversarial Training (DAT) methods are proposed to reduce the domain discrepancy through a domain discriminator (Arjovsky et al., 2017; Ganin et al., 2016; Tzeng et al., 2017; Ganin and Lempitsky, 2015). To learn domain-invariant feature representations, DAT employs a gradient reversal layer to maximize the minimal loss of the domain discriminator. However, DAT suffers from a gradient vanishing problem when the domain discriminator can tell apart the two domains easily, and Wasserstein distance based methods are proposed to enable a more stable training (Shen et al., 2018; Arjovsky et al., 2017; Shah et al., 2018).

6 Future Work and Limitations

Bias and Fairness Similar to other existing NLP models, our models also suffer from bias towards certain demographic attributes (Kusner et al., 2017; Garg et al., 2019; Dwork et al., 2012). For instance, the model may give higher `hate` predictions if the input contains `gay` and higher `sexual` predictions if the input contains `her`. This is because we use data from the Internet, and social bias may present explicitly or implicitly in the training datasets. We tried mitigation methods such as creating a balanced synthetic dataset with templates but could not fully eliminate the issue. In the future, we will continue following related research and improve the fairness of our models.

Data Augmentation We plan to investigate more data augmentation methods to boost the training dataset. Although our current training dataset naturally includes misspelled words and incorrect grammar as some of it is user-generated content, it is valuable to experiment with data augmentation to improve lexicon robustness (Wei and Zou, 2019; Kobayashi, 2018; Zhang et al., 2021) and the generalizability of the model (Guo et al., 2019; Shen et al., 2020; Gao et al., 2021), especially when working with the changing distribution of real-world data.

Better Multilingual Support Only about 5% of the samples are non-English in our training set. As the vast majority of OpenAI API traffic is in English, we have not yet rigorously evaluated or optimized performance on non-English text. Multilingual toxic content classification (Aluru et al., 2020; Wang and Banko, 2021; Lees et al., 2022) would require more non-English training data and may need additional changes on tokenization or model architecture.

Red-teaming at scale Red-teaming is an effective way to find unknown failure cases for the model. Currently we do internal red-teaming with each new model version, which is not a scalable approach. In the future, we plan to set up a pipeline for model red-teaming similar to the one we have for labeling production traffic. We plan to use a specialized interface inspired by Kiela et al. (2021);

Ziegler et al. (2022) to improve the efficiency of the red-teamers.

More Active Learning Experiments Our current active learning strategy to select high-value data for labeling is quite simple. For example, we did not explore diversity sampling due to computational restriction. Onward we plan to run more rigorous experiments comparing the performance of different active learning strategies, as well as more sophisticated strategies, incorporating both uncertainty and diversity sampling.

7 Broader Impacts

Content moderation classifiers have many uses; For example, when paired with fair and robust enforcement practices, they have the potential to reduce certain instances of misuse [6] by ensuring continued moderation of the data generated by and fed into the LM at scale. They also enable filtration of datasets at scale, which may be used to train language models (Welbl et al., 2021) and allow for ease of evaluating language models (Gehman et al., 2020). Longer-term, content moderation classifiers can be used as a way to ensure high-stakes reliability in very-capable AI systems (Ziegler et al., 2022)—a critical necessity for enabling the deployment of those systems in certain domains.

While this underscores the importance of the undesired content classifiers, all classifiers rest on certain assumptions and decisions that may present vulnerabilities or make them inappropriate for certain use cases or types of text. Additionally, these tools can suffer from problematic biases, such as having a high number of false positives targeting speech discussing groups that are frequently the target of hate. (Garg et al., 2019)

The following sections discuss the normative and subjective questions on which these classifiers rest and explore the challenges they present.

7.1 Challenges of Taxonomy Design

We take care to design our taxonomy to reflect generalizable viewpoints. However, much of our data is drawn from a US-centric context and the taxonomy was designed to best fit this data. Additionally, while we have designed our taxonomy to be as comprehensive as possible, it would still be useful for future researchers to add and update the categories based on their own use cases and deployment contexts. Given the sensitive nature of the task, we also encourage the use of this taxonomy in concert with other resources, as it is not possible for any one taxonomy to capture every viewpoint and context.

We hope that this work will encourage further discussion and debate around the principles and values that underpin content moderation.

7.2 Annotator Viewpoints and Disagreement

It is commonly agreed that the annotation of toxic language is an inherently subjective task, and annotators' interpretations may be influenced by their personal and cultural backgrounds, including lived experiences, values and demographic factors.

For example, Waseem and Hovy (2016) found that feminist and anti-racist activists systematically disagree with crowd workers on their hate speech annotations. In their study, agreement between the authors, amateurs and expert annotators is low (14%), most often because in many instances where the authors had identified hate speech, annotators do not.

By necessity, incorporating diverse viewpoints invites disagreement on annotation labels. Much of the computer science literature focuses on eliminating inter-annotator disagreements, most often via deliberation or majority vote. However, in the case of data from or about marginalized populations, disagreement may not be negative, but rather a meaningful signal. An adverse effect of majority vote in such cases is limiting representation of minority perspectives in data (Prabhakaran et al., 2021), potentially reinforcing societal disparities and harms. Moreover, analyzing disagreements may lead to a better understanding of the domain of application (Patton et al., 2018).

In their study, rather than aggregating, Davani et al. (2021) preserve annotator disagreements, which they note could reflect useful and nuanced information about the uncertainty of a sample's membership to a class. Indeed, they demonstrate that their approach yields the same or better performance than similar approaches with aggregated labels, while retaining the ability to estimate uncertainty in predictions that correlate with real-life annotator disagreements.

Moreover, resolving disagreement via majority vote may be at odds with preserving minority opinions in subjective tasks. Alm (2011) argues that

[6] misuse may be defined as uses of the model that the moderating body does not want to allow, e.g., generation of hateful content

achieving a single real "ground truth" label is impossible and is not essential in subjective tasks, and calls for finding ways to model subjective interpretations of annotators, rather than seeking to reduce the variability in annotations.

7.3 Annotator Selection and Welfare

We are committed to ensuring that our labeling tasks are managed in a considerate and ethical manner. All labelers are made aware of the risks and potential harms of working with sensitive data, particularly in the context of potentially encountering their own traumatic experiences while labeling, and we provide them with access to mental health and wellness resources.

7.4 Summary of Broader Impacts Discussion

Content moderation classifiers are one key tool that empowers developers of language models at every stage of the model development and deployment process, from working with large-scale datasets, to testing out models, to deploying the models to many users. However, as we have observed above, there are a range of normative and subjective decisions made throughout the development process of building these classifiers from designing taxonomies to labeling data. Given the nature of these tools, these decisions are sometimes distilled down bluntly and do not enable capturing the nuances that the moderation decision may warrant. This highlights some inherent limitations of classifiers, using automated tools for content moderation, and point to the importance of their robust testing to ensure suitability for each specific use that they may be deployed in.

8 Conclusion

Building high-quality undesired content detection systems in the real world is a difficult challenge that requires the incorporation of multiple methods. A good content taxonomy is the foundation for problem scoping and data collection. A reliable data pipeline is needed to guarantee high data quality and to handle distribution shift. We show that in cases where certain target content occurs rarely, an active learning sampling strategy leads to much better model performance. Additionally, we argue that good operational aspects of the labeling pipeline are essential for ensuring high data quality. And we show that model performance can further be improved through the use of curated synthetic data and semi-supervised learning.

As large generative language models become more and more prevalent, it becomes increasingly important to develop ways of controlling and guiding their outputs. The goal of this work has been to demonstrate one way of implementing such control by way of building content detection models. We are looking forward to further refinement of our approach in the future, as well as progress in other methods of controlling and aligning generative model outputs.

9 Acknowledgments

This work would not have been possible without the contributions of data workers. We greatly appreciate their work handling sensitive content and helping us build better automated systems to make content moderation work less demanding of human labor.

We also thank Miles Brundage, Raf Jakubanis, Gretchen Krueger and ... for feedback on the paper.

References

Cecilia Ovesdotter Alm. 2011. Subjective natural language problems: Motivations, applications, characterizations, and implications. In *ACL (2)*, pages 107–112.

Sai Saketh Aluru, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee. 2020. Deep learning models for multilingual hate speech detection. *arXiv preprint arXiv:2004.06465*.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7383–7390.

OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. 2020. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650. Association for Computational Linguistics.

Gonzalo Molpeceres Barrientos, Rocío Alaiz-Rodríguez, Víctor González-Castro, and Andrew C Parnell. 2020. Machine learning techniques for the detection of inappropriate erotic content in text. *International Journal of Computational Intelligence Systems*, 13(1):591–603.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando C Pereira, and Jennifer Wortman Vaughan. 2009. A theory of learning from different domains. *Machine Learning*, 79:151–175.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando C Pereira. 2006. Analysis of representations for domain adaptation. In *NIPS*.

John Blitzer, Ryan T. McDonald, and Fernando C Pereira. 2006. Domain adaptation with structural correspondence learning. In *EMNLP*.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion Proceedings of The 2019 World Wide Web Conference*, WWW '19, page 491–500, New York, NY, USA. Association for Computing Machinery.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Aaron Daniel Cohen, Adam Roberts, Alejandra Molina, Alena Butryna, Alicia Jin, Apoorv Kulshreshtha, Ben Hutchinson, Ben Zevenbergen, Blaise Hilary Aguera-Arcas, Chung ching Chang, Claire Cui, Cosmo Du, Daniel De Freitas Adiwardana, Dehao Chen, Dmitry (Dima) Lepikhin, Ed H. Chi, Erin Hoffman-John, Heng-Tze Cheng, Hongrae Lee, Igor Krivokon, James Qin, Jamie Hall, Joe Fenton, Johnny Soraker, Kathy Meier-Hellstern, Kristen Olson, Lora Mois Aroyo, Maarten Paul Bosma, Marc Joseph Pickett, Marcelo Amorim Menegali, Marian Croak, Mark Díaz, Matthew Lamm, Maxim Krikun, Meredith Ringel Morris, Noam Shazeer, Quoc V. Le, Rachel Bernstein, Ravi Rajakumar, Ray Kurzweil, Romal Thoppilan, Steven Zheng, Taylor Bos, Toju Duke, Tulsee Doshi, Vinodkumar Prabhakaran, Will Rusch, YaGuang Li, Yanping Huang, Yanqi Zhou, Yuanzhong Xu, and Zhifeng Chen. 2022. Lamda: Language models for dialog applications. In *arXiv*.

Aron Culotta and Andrew McCallum. 2005. Reducing labeling effort for structured prediction tasks. In *AAAI*, volume 5, pages 746–751.

Ido Dagan and Sean P Engelson. 1995. Committee-based sampling for training probabilistic classifiers. In *Machine Learning Proceedings 1995*, pages 150–157. Elsevier.

Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. 2021. Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *CoRR*, abs/2110.05719.

Thomas Davidson, Debasmita Bhattacharya, and Ingmar Weber. 2019. Racial bias in hate speech and abusive language detection datasets. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 25–35, Florence, Italy. Association for Computational Linguistics.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515.

Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. Hate Speech Dataset from a White Supremacy Forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Emily Dinan, Samuel Humeau, Bharath Chintagunta, and Jason Weston. 2019. Build it break it fix it for dialogue safety: Robustness from adversarial human attack. *arXiv preprint arXiv:1908.06083*.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, page 214–226, New York, NY, USA. Association for Computing Machinery.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium. Association for Computational Linguistics.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pages 1183–1192. PMLR.

Yaroslav Ganin and Victor S. Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. *ArXiv*, abs/1409.7495.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Sahaj Garg, Vincent Perot, Nicole Limtiaco, Ankur Taly, Ed H. Chi, and Alex Beutel. 2019. Counterfactual fairness in text classification through robustness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '19, page 219–226, New York, NY, USA. Association for Computing Machinery.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *NIPS*.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572.

Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. *arXiv preprint arXiv:2203.09509*.

Steven CH Hoi, Rong Jin, Jianke Zhu, and Michael R Lyu. 2006. Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd international conference on Machine learning*, pages 417–424.

Jigsaw. a. Perspective api. `https://www.perspectiveapi.com/`. Accessed: 2022-06-15.

Jigsaw. b. Toxic comment classification challenge. `https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/overview`. Accessed: 2022-06-15.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245*.

Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual fairness. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4066–4076. Curran Associates, Inc.

Irene Kwok and Yuzhou Wang. 2013. Locate the hate: Detecting tweets against blacks. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, AAAI'13, page 1621–1622. AAAI Press.

Alyssa Lees, Vinh Q Tran, Yi Tay, Jeffrey Sorensen, Jai Gupta, Donald Metzler, and Lucy Vasserman. 2022. A new generation of perspective api: Efficient multilingual character-level transformers. *arXiv preprint arXiv:2202.11176*.

David D Lewis and Jason Catlett. 1994. Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994*, pages 148–156. Elsevier.

David D Lewis and William A Gale. 1994. A sequential algorithm for training text classifiers. In *SIGIR'94*, pages 3–12. Springer.

Tong Luo, Kurt Kramer, Dmitry B Goldgof, Lawrence O Hall, Scott Samson, Andrew Remsen, Thomas Hopkins, and David Cohn. 2005. Active learning to recognize multiple types of plankton. *Journal of Machine Learning Research*, 6(4).

Y. Mansour, Mehryar Mohri, and Afshin Rostamizadeh. 2008. Domain adaptation with multiple sources. In *NIPS*.

Andrew McCallum and Kamal Nigam. 1998. Employing em and pool-based active learning for text classification. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, page 350–358, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Pamela Mishkin, Lama Ahmad, Miles Brundage, Gretchen Krueger, and Girish Sastry. 2022. Dall·e 2 preview - risks and limitations.

Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios Tsoumakas. 2020. Ethos: an online hate speech detection dataset. *arXiv preprint arXiv:2006.08328*.

Hieu T Nguyen and Arnold Smeulders. 2004. Active learning using pre-clustering. In *Proceedings of the twenty-first international conference on Machine learning*.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, page 145–153, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

PAI. 2021. `https://partnershiponai.org/paper/responsible-sourcing-considerations/`.

Desmond Patton, Philipp Blandfort, William Frey, Michael Gaskell, and Svebor Karaman. 2018. Annotating twitter data from vulnerable populations: Evaluating disagreement between domain experts and graduate student annotators.

John Pavlopoulos, Jeffrey Sorensen, Lucas Dixon, Nithum Thain, and Ion Androutsopoulos. 2020. Toxicity detection: Does context really matter? *arXiv preprint arXiv:2006.00998*.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*.

Vinodkumar Prabhakaran, Aida Mostafazadeh Davani, and Mark Diaz. 2021. On releasing annotator-level labels and information in datasets. In *Proceedings of The Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alan Ramponi and Barbara Plank. 2020. Neural unsupervised domain adaptation in nlp—a survey. *ArXiv*, abs/2006.00632.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2020. A large-scale semi-supervised dataset for offensive language identification. *arXiv preprint arXiv:2004.14454*.

Paul Röttger, Bertie Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts, and Janet Pierrehumbert. 2021. HateCheck: Functional tests for hate speech detection models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 41–58, Online. Association for Computational Linguistics.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678, Florence, Italy. Association for Computational Linguistics.

Tobias Scheffer, Christian Decomain, and Stefan Wrobel. 2001. Active hidden markov models for information extraction. In *International Symposium on Intelligent Data Analysis*, pages 309–318. Springer.

Timo Schick and Hinrich Schütze. 2021. Generating datasets with pretrained language models. *arXiv preprint arXiv:2104.07540*.

Sebastian Schmidt, Qing Rao, Julian Tatsch, and Alois Knoll. 2020. Advanced active learning strategies for object detection. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 871–876. IEEE.

Greg Schohn and David Cohn. 2000. Less is more: Active learning with support vector machines. In *ICML*.

Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *proceedings of the 2008 conference on empirical methods in natural language processing*, pages 1070–1079.

H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. 1992. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294.

Darsh Shah, Tao Lei, Alessandro Moschitti, Salvatore Romeo, and Preslav Nakov. 2018. Adversarial domain adaptation for duplicate question detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1056–1063, Brussels, Belgium. Association for Computational Linguistics.

Dinghan Shen, Mingzhi Zheng, Yelong Shen, Yanru Qu, and Weizhu Chen. 2020. A simple but tough-to-beat data augmentation approach for natural language understanding and generation. *arXiv preprint arXiv:2009.13818*.

Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. 2018. Wasserstein distance guided representation learning for domain adaptation. In *AAAI*.

Xuehua Shen and ChengXiang Zhai. 2005. Active feedback in ad hoc information retrieval. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 59–66.

Aditya Siddhant and Zachary C. Lipton. 2018. Deep Bayesian active learning for natural language processing: Results of a large-scale empirical study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2904–2909, Brussels, Belgium. Association for Computational Linguistics.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. 2018. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *CoRR*, abs/1312.6199.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2962–2971.

Bertie Vidgen and Leon Derczynski. 2020. Directions in abusive language training data, a systematic review: Garbage in, garbage out. *Plos one*, 15(12):e0243300.

Bertie Vidgen, Alex Harris, Dong Nguyen, Rebekah Tromble, Scott Hale, and Helen Margetts. 2019. Challenges and frontiers in abusive content detection. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 80–93, Florence, Italy. Association for Computational Linguistics.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2020. Learning from the worst: Dynamically generated datasets to improve online hate detection. *arXiv preprint arXiv:2012.15761*.

Cindy Wang and Michele Banko. 2021. Practical transformer-based multilingual text classification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pages 121–129, Online. Association for Computational Linguistics.

Zirui Wang, Adams Wei Yu, Orhan Firat, and Yuan Cao. 2021. Towards zero-label language learning. *arXiv preprint arXiv:2109.09193*.

Zeerak Waseem. 2016. Are you a racist or am I seeing things? annotator influence on hate speech detection on Twitter. In *Proceedings of the First Workshop on NLP and Computational Social Science*, pages 138–142, Austin, Texas. Association for Computational Linguistics.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6383–6389, Hong Kong, China. Association for Computational Linguistics.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. 2021. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*.

Karl R. Weiss, Taghi M. Khoshgoftaar, and Dingding Wang. 2016. A survey of transfer learning. *Journal of Big Data*, 3:1–40.

Johannes Welbl, Amelia Glaese, Jonathan Uesato, Sumanth Dathathri, John Mellor, Lisa Anne Hendricks, Kirsty Anderson, Pushmeet Kohli, Ben Coppin, and Po-Sen Huang. 2021. Challenges in detoxifying language models. *arXiv preprint arXiv:2109.07445*.

Zuobing Xu, Ram Akella, and Yi Zhang. 2007. Incorporating diversity and density in active learning for relevance feedback. In *European Conference on Information Retrieval*, pages 246–257. Springer.

Wenjie Yin and Arkaitz Zubiaga. 2021. Towards generalisable hate speech detection: a review on obstacles and solutions. *PeerJ Computer Science*, 7:e598.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the type and target of offensive posts in social media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1415–1420. Association for Computational Linguistics.

Xiangkai Zeng, Sarthak Garg, Rajen Chatterjee, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Empirical evaluation of active learning techniques for neural mt. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 84–93.

Chong Zhang, Jieyu Zhao, Huan Zhang, Kai-Wei Chang, and Cho-Jui Hsieh. 2021. Double perturbation: On the robustness of robustness and counterfactual bias evaluation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, pages 3899–3916, Online. Association for Computational Linguistics.

Daniel M Ziegler, Seraphina Nix, Lawrence Chan, Tim Bauman, Peter Schmidt-Nielsen, Tao Lin, Adam Scherlis, Noa Nabeshima, Ben Weinstein-Raun, Daniel de Haas, et al. 2022. Adversarial training for high-stakes reliability. *arXiv preprint arXiv:2205.01663*.

A Experiment Details

Table 6 presents how we map model taxonomies into labels of different evaluation datasets. Some of the mappings are only approximation. For example, Perspective defines "threat" as "Describes an intention to inflict pain, injury, or violence against an individual or group.", not including graphic violence, so not a perfect match for our "violence" category. Either or our taxonomy has a good match for "toxic", "severe toxic", or "offensive.

Our Evaluation Set We are aware that about 4% of our evaluation samples are in non-English. Perspective API call takes the language as an input parameter, but multilingual is not supported for several attributes. We instead use "en" for all the calls.

Jigsaw Jigsaw dataset is pretty large and we include about half of it into our training set to resolve the cold-start problem. Among the rest half, we sampled 5000 examples for evaluation.

TweetEval We take the TweetEval (Barbieri et al., 2020) test datasets[7] on "hate" and "offensive". There are in total 2970 samples in the hate task test set and 860 in the offensive one.

Stormfront We use the test dataset of de Gibert et al. (2018)[8], containing 478 samples.

Reddit We downsampled 5000 examples from the "RS_201501" snapshot of Reddit pushshift datasets[9] and assigned noisy binary label to each example on whether it contains sexual content according to the subreddits as listed in Barrientos et al. (2020).

[7] https://github.com/cardiffnlp/tweeteval/tree/main/datasets

[8] https://github.com/Vicomtech/hate-speech-dataset

[9] https://files.pushshift.io/reddit/submissions/

Taxonomy		Perspective	Ours	
Ours	Sexual	max(sexually_explicit, profanity, flirtation)	sexual	
	Hate	identity_attack	hate	
	Violence	threat	violence	
	Harassment	max(toxicity, severe_toxicity, insult, threat)	harassment	
	Sexual/minors	-	child_exploitation	sexual/minors
Jigsaw	Toxic	toxicity	harassment	
	Obscene	max(sexually_explicit, profanity)	sexual	
	Threat	threat	violence	
	Insult	insult	harassment, hate	
	Identity hate	identity_attack	hate	
TweetEval	Hate	identity_attack	hate	
	Offensive	max(toxicity, severe_toxicity, threat, insult, identity_attack)	harassment	
Stormfront	Hate	identity_attack	hate	
Reddit	Sexual	max(sexually_explicit, profanity, flirtation)	sexual	sexual

Table 6: How taxonomies of different APIs get mapped into labels of various evaluation datasets.